My Testing Approach

I will test how the Checkout flow behaves when many people use it at the same time.

1. Normal Usage Test

Goal: Check if checkout works smoothly when a normal number of users shop.
Steps tested:

- Open homepage
- Browse products
- Add a product to cart
- Go to checkout
- Place order

What I will measure:

- How fast each step loads
- Whether any errors appear
- Whether the site can handle the daily traffic comfortably

2. Heavy Traffic Test

Goal: See how the website behaves when *a* lot of people try to shop at the same time (like a sale).
What I will check:

- Does the checkout become slow?
- Does the website show errors?
- Does it stop responding?

This helps identify the maximum load the site can handle.

3. Sudden Traffic Test

Goal: Simulate a sudden jump in users (flash sale).
What I will check:

- Does the site freeze?
- Does checkout fail suddenly?

4. Long Duration Test

Goal: Keep users shopping continuously for several hours.
What I will check:

- Does the site become slower over time?
- Does the server memory increase continuously?

- Does anything crash after long usage?

This helps identify memory leaks or performance degradation.

5. Same Product Test

Goal: Test what happens when many users try to buy the same product at once.
What I will check:

- Does the system oversell?
- Does it handle stock correctly?
- Do users get proper messages when stock runs out?

Key things I will measure

- How fast pages load
- Whether any errors happen
- Whether orders are completed successfully
- How stable the website remains under different conditions

Summary

I would performance test the Checkout flow because it is business critical scenario, and I would test it under normal, heavy, sudden, and long duration traffic to ensure speed, stability, and reliability.